A Framework for Binding and Retrieving Class-Specific Information to and from Image Patterns using Correlation Filters

Vishnu Naresh Boddeti, *Student Member, IEEE,* and B.V.K Vijaya Kumar, *Fellow, IEEE*

Abstract—We describe a template-based framework to bind class-specific information to a set of image patterns and retrieve that information by matching the template to a query pattern of the same class. This is done by mapping the class-specific information to a set of spatial translations which are applied to the set of image patterns from which a template is designed taking advantage of the properties of correlation filters. The bound information is retrieved during matching with an authentic query by estimating the spatial translations applied to the images that were used to design the template. In this paper we focus on the problem of binding information to biometric signatures as an application of our framework. Our framework is flexible enough to allow spreading the information to be bound over multiple pattern classes which in the context of biometric key-binding enables multi-class and multi-modal biometric key-binding. We demonstrate the effectiveness of the proposed scheme via extensive numerical results on multiple biometric databases.

Index Terms—Biometric Security, Correlation Filters, Biometric Key-Binding, Face Recognition, Palmprint Recognition

1 Introduction

Among several approaches for image classification, template-based methods, where a single template is designed from a set of gallery images representative of the pattern class, have been popular. Usually these templates are stored in a database along with any class-specific information and classification is done by matching this template to the query pattern. However in some applications (e.g., security related), we want this class-specific information (e.g., a cryptographic key corresponding to that class) to be released only upon a successful match between the template and the query pattern and as such we want the process of information release to be an integral part of matching.

In this paper we describe a framework that allows us to securely bind class-specific information to image patterns and retrieve it only upon a successful match between a query pattern and the corresponding template. A natural application for our framework is biometric authentication (we use this application throughout this paper as an example to evaluate the effectiveness of the proposed framework), where upon successful biometric verification, some class-specific secret information, (e.g., a cryptographic key) is released. Traditional biometric authentication systems release cryptographic keys when the output of the matcher, which compares the template of the claimed class and the biometric signature, indicates an authentic match. However the security of such a system can be compromised by overriding the actual output of the matcher with a fake match score to directly release the cryptographic key. This problem can be mitigated by combining the two tasks of biometric matching and cryptographic key release into a single step. However, the task of binding and accurately retrieving information from templates is challenging since query patterns (e.g., biometric signatures) are usually noisy, either due to their natural variability or noise in the image acquisition process, which can result in errors in the information retrieved from the template.

Since we concentrate on biometric key-binding as the primary application of our framework, we now describe some other desirable properties [1] of the designed templates:

1) ***Revocability***: Just as forgotten or stolen passwords are easy to revoke and new passwords to re-issue, biometric templates should also allow the same.
2) ***Security***: It must be computationally very hard to reverse engineer the information bound to the biometric template without an authentic biometric sample. Also, it should be computationally hard to reverse engineer the raw biometric sample from the template.
3) ***Performance***: In order to ensure that only an authentic query is able to release the key, the error rates i.e., the probability of failure to retrieve the keys for an authentic match and the probability of key retrieval by an impostor match should be low.
4) ***Diversity***: Should have the ability to issue diverse templates, obtained from the same biometric, to be used in multiple applications.

In this paper, we will analyze a biometric key-binding framework that exhibits the above attributes.

The main idea of the proposed method for binding information (interchangeably referred to as key) to im-

- *The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213.*
E-mail: naresh@cmu.edu, kumar@ece.cmu.edu

age patterns is to map this information into parameters for geometric image transformations like rotation, scaling, translation etc. (restricted to spatial translations only in this paper), transform the training images based on these parameters and create a template using the transformed images. During authentication, the query is matched with the template and the image transformation parameters are estimated from the output of the matching process from which the information bound to the template is retrieved.

Given training images $\mathbf{I_1}$, $\mathbf{I_2}$, $\ldots$, $\mathbf{I_M}$ and spatial translation parameters (a_1, a_2) (extracted from the key) the template $\mathbf{f}$ is designed as a weighted sum of the transformed training images with the weights obtained to optimize some appropriate criterion (like tolerance to noise etc.)

$$\mathbf{f} = \sum_i \sum_j \Psi_i (\mathbf{I_i} \ominus \Gamma_j) \tag{1}$$

where $\mathbf{\Psi}_i$ is a data-dependent weighting matrix for each transformed image and $\mathbf{J} = \mathbf{I} \ominus \Gamma$ denotes a projective transformation of an image $\mathbf{I}$ by a projection matrix Γ. During the authentication stage, given only the template $\mathbf{f}$ and an authentic query $\mathbf{y}$, we try to extract the spatial translation parameters by cross-correlating the query and the template, from which the key bound to $\mathbf{I_i}$ can be extracted. The template is designed to facilitate the extraction of the translation parameters from the cross-correlation output.

Building upon the basic idea of cross-correlation for matching a pattern with a template, many advanced correlation filters (CFs) have been developed for shift-invariant object recognition. These filters can be designed to tolerate appearance variability in the image, thereby facilitating stable retrieval of spatial translations. We extend the traditional CF design principles to design templates which while performing the primary task of pattern matching, also have key-binding functionality built into them.

A preliminary version of this work appeared in [2], where we introduced the general framework for binding cryptographic keys to biometric templates. In this paper we present a more robust version of that algorithm which significantly increases the length of the key that can be bound to the template, and improves the failure rate for retrieving the information for an authentic match and the information retrieval rate for an impostor input. Specifically our main contributions in this paper are:

1) **Robust Information Retrieval:** Key retrieval from the template for an authentic match is susceptible to natural variability in biometric signatures. We introduce the use of a sparsity prior while matching the query pattern with the CF for improved robustness.
2) **Probabilistic Decoding:** Instead of thresholding the correlation output to determine the key bound to the template, we propose a maximum-a-posteriori probability (MAP) estimate of the key among candidate keys by a probabilistic mapping of the correlation output thereby improving the key retrieval rate.
3) **Error Correction:** While CFs offer some tolerance to pattern distortions, large distortions in the query pattern degrade the correlation outputs causing errors in the retrieved key. However, an attractive feature of CFs is the graceful degradation of the matching output to noise, occlusions or other distortions in the query. We take advantage of this by using error correction codes (ECC) to handle errors in the retrieved key, leading to better key retrieval rates.
4) **Extensions:** Finally we show how the same key binding framework can be used to spread secret information over multiple pattern classes by designing a single template. This allows us to spread the key over multiple users or multiple biometric modalities thereby enabling multi-user or multi-modal or both multi-user and multi-modal biometric key-binding.

The remainder of the paper is organized as follows. We briefly review recent literature for binding information to biometric signatures in Section 2, following which we review the CF design formulation used in this paper and its extension to bind keys to the filter in Section 3. The key-binding framework is described in Section 4 and in Section 5 we analyze how secure the biometric and the information bound to the template are. Experimental results are presented in Section 6 followed by the multi-class and multi-modal extensions of our framework in Section 7. Finally we conclude in Section 8.

2 Related Work

Biometric authentication being a natural application where one would like to securely bind and retrieve information to and from image patterns, there have been many attempts to address this problem. The goal of all proposed methods is to account for the natural variability in biometric signatures while producing stable outputs. We briefly describe some of the main ideas proposed in the literature.

Soutar et al. [3][4][5] proposed a correlation-based key binding algorithm for fingerprint-based authentication. They design a CF from representative training fingerprint images, set the CF magnitude to one for all spatial frequencies while adding to the filter a random phase array (generated from a password or another biometric modality for example). Key-binding is done by linking the key to the binarized correlation plane. Although we use CFs in our framework, our work differs significantly from this early work in both the CF design and the key-binding algorithm.

Another approach is to extract features that are relatively stable to appearance variations from the training patterns, which are then matched with the features extracted from a query sample. Juels and Sudan [6] proposed a scheme where the secret information is embedded in a fuzzy vault V_F with a set of features S_A. This secret information can be recovered by presenting another set of features S_B which is close to S_A. This scheme has been evaluated for fingerprint-based recognition [7] and iris recognition [8] using a polynomial for binding the key to the biometric features. Hao et al. [9] encode binary keys to iriscodes by an XOR operation between the two while using Hadamard and Reed-Solomon codes to account for variability in iriscodes.

Sutcu et al. [10][11] perform syndrome encoding using a low density parity check (LDPC) code for fingerprint-based matching. While fuzzy vaults offer some tolerance to biometric appearance variability, they have some limitations [12][13] (for example, the secret information can be compromised if the same biometric data is reused in multiple systems, attacks based on statistical analysis of the structure in biometrics are possible and the original template can be obtained from the query pattern knowing that it is an authentic match etc.), some of which are being addressed [14]. A scheme where a hash is generated from the features and matching is done by comparing the hashed features was proposed in [15].

3 Correlation Filters

Since our key-binding framework presented here heavily uses CFs, we provide a brief review of them. More details can be found elsewhere [16]. A CF is a spatial-frequency array (equivalently, a template in the image domain) that is specifically designed from a set of training patterns that are representative of a particular class (a class could be a single image or a single individual or even a group of individuals). This template is compared to a query image by obtaining the cross-correlation as a function of relative shift between the template and the query. For computational efficiency this is computed in the frequency domain (u,v), i.e.,

$$C(u,v) = I^*(u,v)F(u,v) \tag{2}$$

where $I(u,v)$ is the 2D Fourier transform (FT) of the query pattern and $F(u,v)$ is the CF (i.e., 2D FT of the template) and $C(u,v)$ is the 2D FT of the correlation output $c(x,y)$ with $*$ denoting the complex conjugate. Since the images and their FTs are discrete-indexed, FT here refers to the discrete Fourier transform (DFT) which is implemented via the Fast Fourier Transform algorithm (FFT). The CFs are usually designed to give a sharp peak at the center of the correlation output plane $c(x,y)$ for a centered authentic query pattern and no such peak for an impostor.

3.1 Multi-Peak Correlation Filters

One of the main differences between CFs and other classifiers is that, CFs are designed to not only discriminate between different object classes but to also produce sharp peaks in the correlation output which aids in precisely localizing the target in a scene. This is achieved by minimizing the Average Correlation Energy (ACE), along with some regularization, which serves as a proxy loss function to optimize for peak sharpness (for notational ease expressions are given for 1-D signals). For N_1 training images we have,

$$\min_{\mathbf{f}} \sum_{i=1}^{N_1} \|\mathbf{x_i} \otimes \mathbf{f} - \mathbf{g_i}\|_2^2 + \lambda \|\mathbf{f}\|_2^2 \tag{3}$$

where $\otimes$ denotes the cross-correlation operation, $\mathbf{x_i}$ denotes the $i-$th image, $\mathbf{f}$ is the CF template and $\mathbf{g_i}$ is the desired correlation output for the $i-$th image and λ is the regularization parameter. This optimization problem can be solved very efficiently in the frequency domain where the objective function has the following closed form expression,

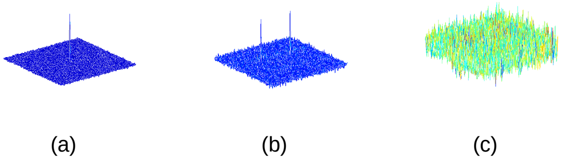


Fig. 1: Correlation Filter Target: (a) traditional correlation output for authentic input with peak at center, (b) multi-peak correlation output for authentic input, (c) correlation output for an impostor input.

$$\min_{\hat{\mathbf{f}}} \sum_{i=1}^{N_1} \hat{\mathbf{f}}^\dagger \hat{\mathbf{X}}_\mathbf{i} \hat{\mathbf{X}}_\mathbf{i}^\dagger \hat{\mathbf{f}} - 2 \sum_{i=1}^{N_1} \hat{\mathbf{g}}_\mathbf{i}^\dagger \hat{\mathbf{X}}_\mathbf{i}^\dagger \hat{\mathbf{f}} + \hat{\mathbf{g}}^\dagger \hat{\mathbf{g}} + \lambda \hat{\mathbf{f}}^\dagger \hat{\mathbf{f}} \tag{4}$$

where $\hat{\mathbf{x}}$ denotes the Fourier transform of $\mathbf{x}$ and $\hat{\mathbf{X}}$ denotes the diagonal matrix whose diagonal entries are the elements of $\hat{\mathbf{x}}$ and $\dagger$ denotes conjugate transpose. Solving the above optimization problem results in the following closed form expression for the CF,

$$\hat{\mathbf{f}} = \left[\lambda \mathbf{I} + \sum_{i=1}^{N_1} \hat{\mathbf{X}}_\mathbf{i} \hat{\mathbf{X}}_\mathbf{i}^\dagger \right]^{-1} \left[\sum_{i=1}^{N_1} \hat{\mathbf{X}}_i \hat{\mathbf{g}}_i \right] \tag{5}$$

where $\mathbf{I}$ is the identity matrix. Depending on the choice of the ideal correlation plane $\mathbf{g}$, the solution to the above optimization lead to the common unconstrained CF designs like Unconstrained Minimum Average Correlation Energy (UMACE) filter [17], Unconstrained Optimal Trade-Off Synthetic Discriminant Function (UOTSDF) filter [18], Maximum Average Correlation Height (MACH) filter [18] and Minimum Output Sum of Squared Error (MOSSE) filter [19]. The key-binding application in our work is enabled by an appropriate choice of the ideal correlation plane $\mathbf{g}$. Fig. 1 shows an illustration of the difference between the output for a traditional CF and the multi-constrained CF used to bind information to the training set. It must be noted that, if the query image is one of the training images with an unknown shift (dx,dy), then while the absolute locations of the peaks will shift by (dx,dy), the relative displacements between these multiple peaks will not change when using good quality images. This property together with the flexibility in designing the filter is exploited to bind the key with the biometric signature.

4 Proposed Framework

The main idea behind our approach is to use CFs for the dual purpose of pattern matching and binding class-specific information to the template. During authentication the bound information is released automatically if the query pattern is authentic. We now describe the enrollment and authentication stages of our framework.

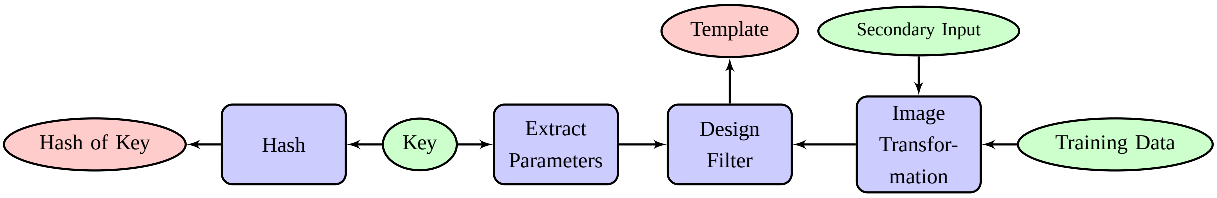


Fig. 2: Block diagram of the training stage. Green denotes the input to the system, blue denotes the processing during training and pink denotes the output of the training phase which is stored in the database. See text for more details.

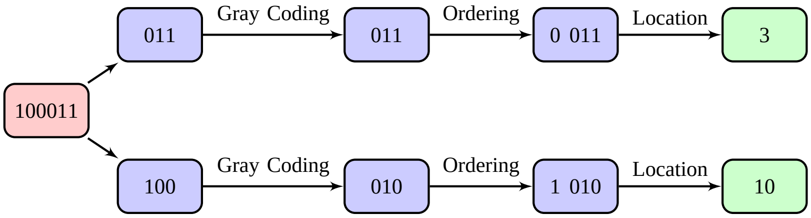


Fig. 3: Example showing the mapping from binary key to peak location. We consider a one dimensional example with 16 possible locations (0-15). We show how a 6 bit key is mapped to a location which is used as an input to the template design algorithm.

4.1 Key-Binding Framework

4.1.1 Enrollment

Fig. 2 shows the block diagram of the training phase of the framework. In this stage the following are required.

- Training images representative of the authentic class. These could be biometric signatures under various conditions (e.g., face images with different lighting, expressions etc.) expected during testing.
- Information or key to be bound to the template.
- A secondary input (e.g., password or pin) is required for additional protection (details in Section 4.1.3).

The information or key to be bound is mapped into locations in the correlation plane, by segmenting the key into smaller segments of appropriate size (See Fig. 3 for an example). Let b be the bit representation of the key with m bits, and q be the total number of segments. The i-th key segment is given as,

$$k_i[n] = b[n] \cdot w[n - \delta_i] \tag{6}$$

$$w[n] = \begin{cases} 1 & \text{if } 0 \leq n \leq r-1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where r is the size of the window and δ_i is the window shift. While other configurations are possible, in this work we only consider non-overlapping windows, $\delta_i = (i-1)r$. Now each segment of the key k_i is mapped to a peak location in the correlation plane, therefore the number of peaks is determined by the length of the key to be bound to the training images. We use these locations to get the ideal correlation output $\mathbf{g}$ in Eq. 3 (*a value of one at the peak locations and zeros everywhere else*) to design the CF for the given class and key pair. For example, in Fig. 3 the ideal correlation plane $\mathbf{g}$ would have peaks (values of one) at locations 3 and 10 and zeros everywhere else. Ideally the multi-peak CF should produce correlation peaks at those locations in response to a centered authentic image. For an un-centered query pattern, the peaks shift globally depending on the unknown relative shift between the query and the training images. Therefore during training, we compute the centroid of the specified peak locations which is stored in the database.

During authentication once the peaks are identified, we shift the centroid of the detected peaks to the location stored in the database. Further, since the key is mapped only to locations in the correlation plane, it can be recovered only up to a permutation if the order of the key segments is not encoded. While other configurations are possible, we address this issue by encoding the ordering along with the key in the filter itself by augmenting the key k_i with the order of that segment. For example, in Fig. 3, we need two peaks (each taking on one of 16 locations) to represent two 3-bit key segments with the extra bit required to encode the ordering of the key segments which reduces the number of key bits that can be represented by a given number of peaks. For images of size d_1xd_2 and a key with m bits, the number of peaks and the window size are related as follows,

$$\begin{aligned} q(\log_2 d_1 + \log_2 d_2) &\geq m + q\log_2 q && (8) \\ r &= \left\lceil \frac{m}{q} \right\rceil && (9) \end{aligned}$$

where $\lceil\rceil$ denotes the ceiling operation and the term $q\log_2 q$ is the number of bits required to encode the peak ordering information. The multi-peak CF is designed with the training images and the constraints obtained from the key as inputs. This template is stored in the database along with the hash value of the key (this is optional) computed using a one-way hash function. Further we also do not allow the locations of the constraints to be within ε (typically 5 or 10 in our experiments) pixels from the boundary of the image.

4.1.2 Authentication

Fig. 4 shows a block diagram of the authentication process. In this stage, the query pattern is provided along with secondary input and a claimed identity in a verification

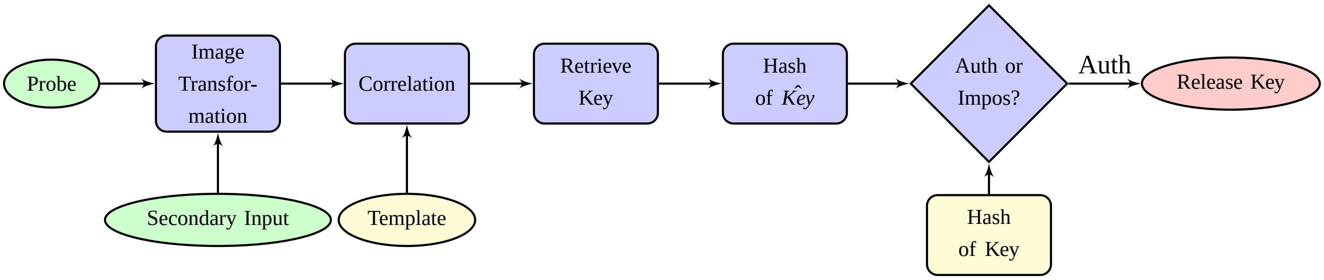


Fig. 4: Block diagram of the testing stage. Green denotes the input from the user, yellow denotes the data stored in the database, blue denotes the processing during the testing stage and pink denotes the output of the system. See text for more details.

scenario. The query is then cross-correlated using the CF corresponding to the claimed identity. If the query is authentic, the resulting correlation should have peaks at the right locations (except for a possible global shift). We then shift the centroid of the detected peaks to the centroid stored in the database. From the new peak locations, the information bound to the template is reconstructed. The correctness of the retrieved key can be confirmed by comparing the stored hash value with the hash value of the recovered information. Only when the two hashes match, is the information released. For an impostor query image, key recovery would fail due to the absence of any sharp peaks in the resulting correlation plane when either the secondary input is incorrect or the query does not belong to the claimed class or when both are incorrect. Note that the hash value is used only to confirm that the recovered information is correct. In situations where such a guarantee is not required or when there are other ways of confirming the accuracy of the retrieved key (e.g., by using the key retrieval confidence measure described in section 4.2.2 to decide the correctness of the retrieved key), the hash of the key is not stored in the database along with the template.

4.1.3 Image Transformation

The main purpose of parametrically transforming the images before being used to design the CF is to prevent information leakage from the template and to help create diverse and cancelable biometric templates from the same biometric samples. Examples of such parametric transforms previously proposed in the literature include non-invertible geometric image transformations [20], convolution with random masks for cancelable CFs [21] and data projection onto random matrices [22]. The “secondary input” is used as the parameter for the image transformations. In this paper we adopted the technique introduced in [21]. Using a user provided password as a seed, a matrix with random entries is generated which is then convolved with the gallery images. The resulting images (see Fig. 5 for an example) are then used to design the CF for that user. During testing, the probe image is supplied along with a password which determines the random mask. This probe is then convolved with the random mask and then correlated with the CF of the corresponding claimed identity. If the probe password matches the password provided by the user at training, the effect of the random mask is canceled out during the correlation operation between the CF and the “transformed” probe image (this is shown in [21]). Therefore, for an authentic match the effect of the random mask is completely neutralized and only the underlying biometric features dominate the matching process. However, for an impostor match the probe random mask does not neutralize the random mask used while training and hence the matching process is dominated by the random masks and therefore improves the non-match performance.

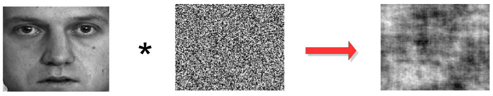


Fig. 5: Transformation by a random convolution kernel.

4.2 Robust Key Retrieval

During decoding the main challenge in recovering the information bound to the template is in robustly identifying the peaks and the peak locations in the correlation plane (from which the key is recovered) since each point in the correlation output is a potential peak. This problem is exacerbated by the presence of false peaks due to variability (we will refer to this variability as noise) in the correlation plane due to variable imaging conditions, pattern variability or sensor noise. This noise leads to errors in identifying which locations in the correlation plane correspond to peaks and also errors in the locations of the detected peaks. We address these errors by first denoising the correlation plane as much as possible following which we use probabilistic estimates to determine the number of peaks and peak locations and finally correcting for any residual errors in the peak locations using error correcting codes.

4.2.1 Exploiting Sparsity for Denoising

We address the noise in the correlation plane based on the following observation. For an authentic match the resulting correlation plane would have only a few high values (peaks) and most of the correlation plane would have low values (noise) i.e., the ideal correlation plane would be highly sparse. Therefore, instead of reconstructing the correlation plane by an inverse FT (which corresponds to minimizing the l_2 norm of the correlation output), reconstructing the correlation plane by minimizing the l_0 norm of the resulting correlation output will result in a less noisy correlation plane. We remind that $F(u,v)$ and $I(u,v)$ denote the CF and the Fourier transform of the test image where u and v denote the 2-D spatial frequencies. Let $\tilde{c}(x,y)$ and $\tilde{C}(u,v)$ be the ideal target correlation output for an authentic match in the spatial and spatial frequency domains respectively i.e., $\tilde{c}(x,y)$ would have large non-zero values at the desired peak locations derived from the key and zeros everywhere else. We model the correlation plane obtained from the filter $F(u,v)$ and query image (whose FT is $I(u,v)$) as a noisy version of the ideal target correlation plane i.e.,

$$C(u,v) = I^*(u,v)F(u,v) = \tilde{C}(u,v) + \eta(u,v) \tag{10}$$

where η denotes noise. So we would like to recover the $N(=d_1 d_2)$ dimensional correlation output as follows,

$$\hat{\mathbf{c}} = \arg\min_{\Theta} \|\mathbf{C} - \mathbf{A}\Theta\|_2^2 \text{ s.t. } \|\Theta\|_0 \leq T_0 \tag{11}$$

where $\Theta \in \mathbb{R}^N$ is the desired correlation plane, $\mathbf{A} \in \mathbb{C}^{N\text{x}N}$ is the DFT matrix and $\mathbf{C} \in \mathbb{C}^N$ is the Fourier transform of the correlation plane obtained from the designed CF and the query image and $\|\Theta\|_0$ is a pseudo norm which counts the number of non-zero values in the correlation plane Θ which ideally must correspond to the number of peaks. The problem in Eq. 11 is NP-hard but it has been shown that under some conditions [23] replacing the l_0 norm with the l_1 norm will result in the same solution for *sufficiently* sparse [24] correlation outputs, i.e.,

$$\hat{\mathbf{c}} = \arg\min_{\Theta} \|\mathbf{C} - \mathbf{A}\Theta\|_2^2 \text{ s.t. } \|\Theta\|_1 \leq T_1 \tag{12}$$

Further since the desired sparsity T_0 (and T_1) is unknown we solve the following $l_2 - l_1$ problem (called *basis pursuit denoising* [25] - BPDN),

$$\min_{\Theta} \ \frac{1}{2}\|\mathbf{C} - \mathbf{A}\Theta\|_2^2 + \tau\|\Theta\|_1 \tag{13}$$

We use the *Sparse Reconstruction by Separable Approximation* (SpaRSA) [26] algorithm to solve Eq. 13. Depending on the similarity of the query to the training patterns used to design the template, different values of τ result in different levels of denoising in the resulting correlation plane. Since denoising is important for key retrieval, we automatically choose the value of τ that gives the best denoised correlation plane. This is done by solving Eq. 13 for an ascending series of values for τ. Starting from a low value of τ, we check at every iteration if the sparsity of the resulting correlation plane is more than the sparsity in the previous iteration. We continue this until BPDN returns an all zero correlation plane, which happens when the regularization cost ($\tau\|\Theta\|_1$) is greater than the noise in the correlation plane. After this the algorithm returns the solution of the previous iteration, since that is the iteration with the best denoised correlation plane. Experimentally, this simple idea was found to work very well.

4.2.2 Probabilistic Framework for Decoding

Once the correlation plane has been obtained either by solving Eq. 2 or Eq. 13, the peaks in the correlation plane need to be identified in the presence of noise. During authentication if the number of key segments q are known (corresponds to knowing the size of the key) then one can simply pick the q highest peaks in the correlation plane and extract the information from the corresponding peak locations. However, in many scenarios the length of the key is unknown, in fact we show in Section 5 that hiding the length of the key makes the information more secure. In this case every location in the correlation plane is valid, hence there are many possible keys that can be extracted from the correlation plane. One can simply determine if a particular location is a peak or not using a hard threshold on the correlation values. However, the noise level in the correlation plane varies from query to query due to image variabilities rendering a fixed threshold suboptimal in terms of separating the peaks from noise. Therefore we perform a probabilistic analysis of the correlation plane to detect and identify peaks. We extract the most likely key from the correlation plane by estimating the most likely number of peaks in the correlation plane. The key idea is to probabilistically map either a single value or a group of values of the correlation output to a posterior probability value on the existence of a peak or a group of peaks. For an N dimensional correlation plane, let $\mathbf{h}$ be a vector of hypotheses with h_i being the hypothesis that the i-th element is a peak, $\neg h_i$ the hypothesis that the i-th element is not a peak, let $\mathbf{c}$ be the correlation values and c_i be the correlation value at the i-th element.

$$\begin{aligned} q^* &= \arg\max_q p(\mathbf{h}^q|\mathbf{c}) && (14)\\ p(\mathbf{h}^q|\mathbf{c}) &\propto p(\mathbf{c}|\mathbf{h}^q)p(\mathbf{h}^q) && (15)\\ p(\mathbf{c}|\mathbf{h}^q) &= p(\mathbf{c}|\mathbf{h}^q_Q, \neg\mathbf{h}^q_{\tilde{Q}}) && (16) \end{aligned}$$

where $p(\mathbf{h}^q)$ is the probability of q peaks in the correlation plane, $Q = \{i|1 \leq i \leq q\}$, $\tilde{Q} = \{i|q+1 \leq i \leq N\}$, $\mathbf{h}^q$ is the hypothesis that the q highest correlation values are peaks, $\mathbf{h}^q_Q$ is a q dimensional vector of hypothesis corresponding to there being q peaks in the correlation plane. $p(\mathbf{c}|\mathbf{h}^q)$ represents the likelihood of the correlation plane $\mathbf{c}$ given the hypothesis $\mathbf{h}^q$ which is learned and $p(\mathbf{h}^q)$ is the prior on the hypothesis that there are q peaks in the correlation plane. Typically since the dimensionality of $\mathbf{h}$ can be very large, modeling this joint distribution is not feasible. Therefore we make the following simplifying assumption,

$$p(\mathbf{c}|\mathbf{h}^q) = p(\mathbf{c}_Q|\mathbf{h}^q_Q)p(\mathbf{c}_{\tilde{Q}}|\neg\mathbf{h}^q_{\tilde{Q}}) \tag{17}$$

where $\mathbf{c}_Q$ is defined just like h^q_Q. We further assume that, the correlation values which are not peaks have statistical distributions that are identical and independent of each other. This is a reasonable assumption since the correlation values which are not peaks are caused by noise. This is however not true of the correlation values which are peaks since all of them are actually caused by the same image.

$$p(\mathbf{c}|\mathbf{h}^q) = p(\mathbf{c}_Q|\mathbf{h}_Q^q) \prod_{i=q+1}^{N} p(c_i|\neg h_i^q) \tag{18}$$

On observing the empirical distributions (from experiments on a small dataset of face images), we found that a multivariate normal distribution and a univariate normal distribution are reasonable approximations for $p(\mathbf{c}_Q|\mathbf{h}_Q^q)$ and $p(c|\neg h^q)$ respectively where c is the correlation output at location h. Further the prior on there being a peak at h is $p(h^q) = \frac{q}{N}$ since the distribution of the peaks in the correlation plane is uniform. We model the prior distribution for each hypothesis $\mathbf{h}^q$ as a binomial distribution,

$$p(\mathbf{h}^q) = \binom{N}{q} p(h^q)^q p(\neg h^q)^{N-q} \tag{19}$$

where $q \in \{0, 1, \ldots, N\}$. During the learning stage the following parameters for our models are estimated from a separate generic training set.

1) $\mu_{\neg h}$ and $\sigma_{\neg h}$ for $p(c|\neg h^q)$
2) μ_Q and Σ_Q for $p(\mathbf{c}_Q|\mathbf{h}_Q^q)$

From empirical eigenvalue analysis of the covariance matrix Σ_Q, we observed that there is only one dominant eigenvalue. Hence in practice instead of the whole covariance matrix we store the largest eigenvalue and its corresponding eigenvector. Since the size of the covariance matrix gives away the number of correlation peaks, we create a single covariance matrix for the maximum allowable peaks and use the sub-matrix depending on the number of peaks. Further in practice instead of searching over all possible q, we can use a simple heuristic (number of peaks corresponding to maximum change in the correlation values sorted in descending order) to find the most likely number of peaks in the correlation plane and do a local search. Once we find the number of peaks, we find the locations of the corresponding peaks to retrieve the key from the template. The probabilistic treatment of the correlation plane also provides a confidence measure for key retrieval from the correlation plane in scenarios where this might be useful.

4.2.3 Robust Decoding with Error Correction

Due to correlations between neighboring pixels in images and noise in the correlation plane, the peaks in the correlation plane are not very sharp resulting in the peak locations sometimes being off by a couple of pixels (Note that these are errors remaining after accounting for any global shift in the correlation peaks that arises due to relative shifts between the query and training patterns). This results in errors in the key retrieved from the corresponding peak locations. We use ECC to handle such errors. To ensure that the size of the parity bits does not reveal the length of the key, we pad all keys up to a known and pre-determined maximum key length and during decoding the key length is determined by the number of peaks detected in the correlation plane. Further since even a single pixel error in the peak locations may result in multiple bit errors in the recovered key, we apply gray coding on the key which is then used in the template design process. This ensures that the degradation pattern (bit errors) in the retrieved key is proportional to the degradation pattern (errors in peak location) in the correlation plane. During decoding once the peak locations are identified, we decode the gray code to piece together the original key.

We consider two ways of applying ECC, at the segment level for each segment that the key is mapped into and at the key level before mapping it into segments. In the former scenario, one can guarantee to correct every segment up to a selected number of errors and is a more natural choice since errors happen at the segment level. However not every segment is in error, hence in the latter scenario, we can achieve error correction performance comparable to the former scenario but with less redundancy. For both the cases we use binary BCH codes (and shortened binary BCH codes where needed) since they are particularly well suited for our problem. Although we do not pursue it in this paper, the correlation peak values can be used as soft information for soft decision decoding to improve ECC performance, perhaps at the cost of increased computational complexity.

5 Security Analysis

Keeping the key and the biometric signature secure is one of the goals of biometric key-binding. In this section we analyze the security of the information bound to the template and the biometric signature itself.

In the scenario where a secondary input is used to map the image into another space, both the biometric signature and the secondary input have to be correct to accurately retrieve the information which effectively gives us a two-factor authentication scheme. Therefore if the biometric signature is compromised, the security of the hidden key depends on the entropy of the secondary input and also on the quality of the biometric signature stolen by the adversary. If the secondary input is compromised the security of the hidden key depends on how well the adversary estimates the biometric signature. If the hidden key is compromised, the adversary can recreate the signal $\mathbf{x}$ used to design the template up to an image mapping (if a secondary input is used). If the image mapping is however a non-invertible transform as presented in [20], then the original biometric signature cannot be recovered.

5.1 Brute Force Security

In this scenario the adversary has access only to the template $\mathbf{f}$ and the centroid of the peak locations. Hence the adversary can retrieve the key only by guessing it i.e., guessing the number of peaks in the correlation plane and the locations of the peaks subject to constraints on where in the correlation plane the peaks can occur (the adversary will have to guess the locations of only $q-1$ peaks since the centroid of the q peak locations is known). Under the assumption that the adversary knows the number of peaks q in the correlation plane (corresponds to knowing the length of the key), the key space in bits that the adversary has to search over is,

$$L(q) = (q-1)\log_2\left[\frac{(d_1-\varepsilon)(d_2-\varepsilon)}{q}\right] \tag{20}$$

where ε is the size of the correlation output boundary region that is ignored. The $\frac{(d_1-\varepsilon)(d_2-\varepsilon)}{q}$ term quantifies the number of available peak locations for each peak and the number of peaks q for a key of length m is governed by Eq. 9. In the scenario where the adversary has no knowledge of the length of the hidden key (in scenarios where the key length is fixed we can augment the key with random data of arbitrary size which is then encoded in the template), then the key space to search over is,

$$L(q*) = \sum_{q=1}^{q^*}(q-1)\log_2\left[\frac{(d_1-\varepsilon)(d_2-\varepsilon)}{q}\right] \tag{21}$$

where $q*$ is the actual (unknown) number of peaks in the correlation plane given by Eq. 9.

Through the rest of this paper we quantify and report brute force security assuming that the adversary knows the number of peaks in the correlation plane (i.e., key size) and is therefore a conservative estimate. The above expressions however do not account for any decrease in key security when using ECC to correct for errors in the detected peak locations. When using ECC, the security of the key reduces by an amount equal to the error correction capability of the code for a properly chosen code, i.e., for a m bit key, t bit error correction per segment ($2t$ bit error correction per peak for the two coordinates) and q peaks, the security is given by $L(q)-2tq$ for perfect codes. However, BCH codes are not perfect codes (i.e., one can sometimes correct more than t errors up to a maximum of $2t$ errors at the expense of computational complexity), therefore we report the range $[L(q)-4tq, L(q)-2tq]$ for our experiments.

5.2 Dictionary Attack

We now consider a more realistic attack which is based on a dictionary of biometric signatures mined from some data source. It must be noted that a dictionary attack can be successful only if the secondary input is known so as to compensate for the convolution with the random mask. Therefore in addition to the template (CF of a particular user) we also assume that the adversary has access to the secondary input (password in our case). Since correlation is a linear operation any query which is a weighted linear combination of the training images will result in a correlation output with sharp peaks from which the key can be retrieved. The goal of an adversary is to estimate an unknown number of transformations $\Gamma_1,\ldots,\Gamma_{q^*}$ given $\mathbf{f}$ which can be done only by estimating any point on the unknown hyperplane formed by the training images or the subspace on which the training images lie on.

Let the match score be the maximum correlation value, i.e., $max\{\mathbf{f}^{\mathbf{T}}(\mathbf{x}\ominus\Gamma)|\Gamma\in(a_1,a_2), 0\leq a_1\leq d_1, 0\leq a_2\leq d_2\}$ which is convex in $\mathbf{x}$ and non-convex in Γ, then given $\mathbf{f}$ an estimate of the signal $\tilde{\mathbf{x}}$ can be obtained by solving the following optimization problem,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{f}^{\mathbf{T}}(\mathbf{x}\ominus\Gamma) \\ s.t. \quad & \forall\ \Gamma\in(a_1,a_2), 0\leq a_1\leq d_1, 0\leq a_2\leq d_2 \\ & \|\mathbf{x}\|_2 = 1 \end{aligned} \tag{22}$$

This is a convex problem in $\mathbf{x}$ for a given Γ and can be solved to obtain an image $\tilde{\mathbf{x}}$ which maximizes the match score for the given Γ. Since Γ is defined over a finite space, only a finite number of convex problems need to be solved. However this does not guarantee that correlating this solution with the filter will result in peaks at the right locations from which the key bound to the template can be retrieved. In fact in the absence of a model for $\mathbf{x}$, the $\tilde{\mathbf{x}}$ which maximizes the match score is the complex conjugate of the template $\mathbf{f}$ from which the key cannot be retrieved.

However in a more realistic scenario the adversary has access to an image model $\mathbf{x}=\mathbf{D}\lambda$ where $\mathbf{D}\in\mathbb{R}^{Nxt}$ is a dictionary for $\mathbf{x}$ (e.g., eigenfaces for a face database from any publicly available face database.) which helps reduce the search space from $\mathbf{x}\in\mathbb{R}^N$ to $\lambda\in\mathbb{R}^t$. In this scenario the adversary can estimate an image $\tilde{\mathbf{x}}$ which potentially helps retrieve the key bound to the template by solving,

$$\begin{aligned} \max_{\lambda} \quad & \mathbf{f}^{\mathbf{T}}(\mathbf{x}\ominus\Gamma) \\ s.t. \quad & \forall\ \Gamma\in(a_1,a_2), 0\leq a_1\leq d_1, 0\leq a_2\leq d_2 \\ & \|\mathbf{x}\|_2 = 1 \\ & \mathbf{x} = \mathbf{D}\lambda \end{aligned} \tag{23}$$

Again this is a convex problem in $\mathbf{x}$ for a given Γ, in fact we can easily derive a closed form expression for $\tilde{\mathbf{x}}$ in terms of $\mathbf{f}$ and $\mathbf{D}$ for a given Γ. Therefore the adversary can estimate an image $\tilde{\mathbf{x}}$ for a large but finite number of values of Γ and pick the image with the highest correlation value. This estimated image $\tilde{\mathbf{x}}$ can now be used to retrieve the key bound to the corresponding template $\mathbf{f}$.

5.3 Substitution Attacks

We now make a brief mention of a particular kind of attack against which the proposed algorithm is unsuitable. When an attacker has access to a user's secret key he/she can design a new template using his/her own biometric signature and password to masquerade as the said user via a simple substitution attack or a blended substitution attack. However, substitution attacks are possible ***only*** if the attacker has access to the secret key that the biometrics are being used to protect. Such attacks enable the adversary to effectively bypass the whole biometric matching algorithm by simply adding himself/herself into the system to release the secret key. The proposed algorithm is ineffective against such attacks and as such is unsuitable for scenarios where the adversary has access to the secret key that the template is supposed to protect. While templates can be designed to prevent two different users to release the same keys, such techniques might not be scalable and are beyond the scope of this paper. A simpler solution to prevent such blended substitution attacks would be to hash the template or digitally sign the template to ensure that it has not been tampered with. The security of the stored hash of the template is now of paramount importance.

6 Experimental Results

To demonstrate the effectiveness of our scheme, we conduct several experiments where we bind random data to biometric signatures and attempt to retrieve this information by matching against query patterns. Further to demonstrate the generality of our framework, we use two different biometric modalities, faces and palmprints to test our proposed framework. Since biometric key-binding is expected to be used for applications such as access control, e-commerce etc., it is reasonable to expect user cooperation while presenting his/her biometric, therefore in our experiments we consider only illumination and expression variation in the query images. Illumination is one variable that cannot be controlled well even with co-operative users and hence became the impairment variable in this paper. Since binding and retrieving information is the goal of the proposed framework, we report performance in terms of information retrieval failure rate (IRFR) and false class information retrieval rate (FCIRR) i.e., the rate at which a false class query pattern can retrieve the information of another pattern class due to the similarities between the two pattern classes. Ideally we would like FCIRR to be zero, since it is not desirable to have an impostor retrieving the information of some other user. For the sake of comparison we also report the results of standard biometric authentication in the form of Equal Error Rate (EER) and the Rank-1 identification rate without biometric key-binding.

6.1 Databases

1) **CMU PIE:** The CMU-PIE database [27] has images of faces with different poses, expressions and illumination variations. We present results using frontal images of neutral expressions with different illuminations, we use both the PIE-lights and PIE-nolights where ambient lights are on and off respectively. The PIE-lights and PIE-nolights databases consist of 68 classes with 24 images per class and 66 classes with 21 images per class respectively. In our experiments we used 3 images for training and the rest of the images for testing. The 3 training images were chosen such that one has illumination from the left, another has illumination from the front and the third has illumination from the right i.e., indices 3,7 and 16.
2) **CMU Multi PIE:** The CMU Multi-PIE database [28] is an extension of the PIE database. It has a total of 337 subjects. As in PIE, different face poses, expressions and illumination variations were recorded. We present results using frontal images of neutral expressions with different illuminations of which there are over 23000 images. We use 15% of these images for training which corresponds to 3 images per session, one with illumination from the left, one with illumination from the right and the third has illumination from the front among 20 different illumination patterns and use the rest for testing.
3) **PolyU:** The PolyU palmprint database [29] consists of 386 classes with over 7500 images collected over two sessions with about 10 images per session. We randomly chose 15% of the images for training and used the rest for testing.
4) **FRGC:** The Face Recognition Grand Challenge database [30] consists of about 50,000 recordings divided into training and validation partitions. The validation set is further divided into a target set and a query set of 466 people. For our experiments we used a 410 class subset of the target set (after eliminating all classes with fewer than 6 images) with a total of 15,804 images. We randomly chose 25% of the images for training and used the rest for testing.

6.2 Pre-Processing

We process the face databases by first running an eye-detector to get the eye locations which are used to register and size normalize the images. Similarly we process the palmprint images as in [31] to extract and register the palmprint from the full images.

6.3 Experiments

We conducted numerical experiments with images of size 128x128 to bind randomly generated keys of length between 20 (corresponds to 2 peaks) and 770 (corresponds to 80 peaks) bits (images of larger size allow us to bind longer keys since more peaks can be packed in the correlation plane). Each experiment is conducted 10 times since the keys bound to the template are generated randomly and we present results averaged over these runs. Table 1 shows the key size[1], brute force security with no error correction, range of brute force security with 1 bit and with 2 bit error correction per peak for different number of peaks for images of size 128x128 and correlation outputs of size 256x256. As explained in Section 5.1, we report a range for the brute force security with error correction since the codes we chose for error correction are not perfect codes. Due to space constraints we do not report in this paper the results of our experiments with constrained filters (e.g., MACE[17], OTSDF[32] etc.) and other unconstrained filters (e.g., MACH[18] etc.) and results with error correction on the full key instead of error correction on each key segment. However, these results follow a similar trend to the results reported here with the other CFs doing worse than UMACE filter in terms of IRFR.

6.3.1 *Biometric Authentication*

We first present results without binding information to the CFs, i.e., the CFs are designed to produce a single peak

[1] Unfortunately due to additions like ECC and ordering information of the key segments it is difficult to ensure that key sizes are powers of 2. So if one requires keys which are powers of 2, one has to use parts of the bit sequence as a pad.

TABLE 1: Brute Force Security

# of	key Size	Brute Force Security (bits)		
Peaks	(bits)	(No ECC)	(1 bit ECC)	(2 bit ECC)
05	68	53	(33, 43)	(13, 33)
10	126	111	(71, 91)	(31, 71)
20	233	217	(137, 177)	(57, 137)
30	332	314	(194, 254)	(74, 194)
40	427	407	(247, 327)	(87, 247)
50	517	495	(295, 395)	(95, 295)
60	605	581	(341, 461)	(101, 341)
70	690	664	(384, 524)	(104, 384)
80	774	746	(426, 586)	(105, 426)

TABLE 2: Biometric Recognition Rates (in %)

	PIE Lights	PIE Nolights	Multi PIE	PolyU	FRGC
EER	0.07	0.25	2.42	0.23	5.2
IDA	100	99.9	99.3	99.7	93.5

for an authentic match at the center of the correlation plane and no such peak for an impostor. We report the Rank-1 identification accuracy (IDA) as well as the Equal Error Rate (EER) for all the databases used for our experiments in Table 2. The results suggest that the databases used in our experiments are relatively easy (excepting FRGC) for CF based recognition.

6.3.2 Known Key Size

Here we present results when the key size is known (nothing else is known about the key) which corresponds to knowing the number of peaks $q*$ in the correlation plane. Hence we retrieve the key simply from the locations corresponding to the $q*$ highest peaks i.e., the MAP estimate described in Section 4.2.2 is not required to determine the number of peaks. Errors in this case are caused by the peaks being in the wrong locations, which are corrected by ECC. Fig. 6a and Fig. 6b shows plots of IRFR for CMU Multi PIE and PolyU Palmprints respectively as a function of the number of peaks in the correlation plane and the error correction capability going from no ECC to full ECC. Table 3 lists the IRFR for a select number of peaks for the different databases. Key retrieval performance is dependent on the sharpness of the peaks and the quality of the query, so the majority of the errors made with shorter keys are due to the poor quality of the query image while majority of the errors with longer keys are caused by reduced peak sharpness which in turn is caused by increased noise in the correlation plane. As expected we see that the key retrieval failure rates increase with the size of the key. Ideally we would like to maximize both peak sharpness and number of peaks in the correlation plane. Both these requirements cannot be achieved simultaneously since there exists a trade-off between how many peaks one can pack in the correlation plane and the sharpness of the peaks. Error correction (even just 1 bit error correction) helps improve the performance and dramatically in the case of palmprints and FRGC. Since FRGC images exhibit harsher illumination and expression variations compared to CMU MPIE the performance on FRGC is worse compared to CMU MPIE. Most importantly when using the secondary input we observe zero FCIRR with no ECC, with 1 bit ECC and with 2 bit ECC, i.e., no impostor was able to retrieve the key of some other user, since to retrieve the key we not only require the right secondary input (password in our case) but also require a partial match between the template and the query and also require all the peaks to be at the right locations which is unlikely for an impostor input because it is harder to identify the number of peaks due to high levels of noise in the correlation plane. Further even if the number of peaks is correctly determined, the peak locations have errors which are beyond the error correction capability of the codes used. Not using the secondary input does result in non-zero FCIRR although the IRFR remains unaffected. Under this scenario Table 4 lists the FCIRR for different databases as we vary key length and the amount of ECC. While the FCIRR is high for shorter keys, it quickly drops down reaching zero for longer keys. Note that since we use an adaptive threshold instead of a fixed threshold to determine the peaks and peak locations in the correlation plane, there is no trade-off between IRFR and FCIRR.

6.3.3 Unknown Key Size

Here we consider the scenario when the key size is unknown which corresponds to unknown number of peaks in the correlation peaks and we use the MAP estimate described in Section 4.2.2 to determine the number of peaks in the output correlation plane. In this case errors occur either due to the wrong number of peaks being detected or the detected peaks being in the wrong locations or both. When the number of peaks detected is wrong, the hash of the retrieved key and the original key do not match, so these errors are simply added to the errors remaining after error correction. Fig. 6c and Fig. 6d shows plots of IRFR for CMU MPIE and PolyU Palmprints respectively as a function of the number of peaks in the correlation plane and the error correction capability going from no ECC to full ECC. Table 5 lists the IRFR for a select number of peaks. The key retrieval performance is worse compared to the scenario where the key size is known as one would expect. However this should also make it harder for an impostor to retrieve somebody else's key as is evident from FCIRR (see Table 6) when not using the secondary input. It must be noted that FCIRR is still zero when using the secondary mask to convolve the images with the random mask.

6.3.4 Dictionary Attack

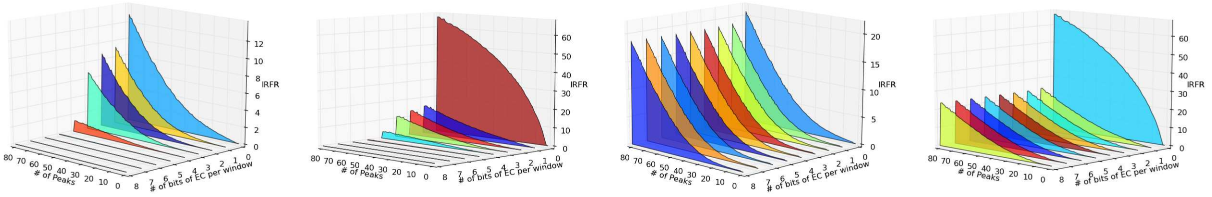
From the perspective of an adversary, we also conducted experiments to retrieve keys bound to the templates of subjects in the CMU PIE-nolights database via a dictionary attack. We consider two scenarios, one where we design the multi-peak CFs using the raw training images and the other where the CFs are designed using training images that are convolved with a random mask (obtained via a password). In each of these cases we use three different dictionaries, DICTA: all images of CMU PIE-nolights, DICTB: all images of CMU PIE-lights and DICTC: 5000 randomly selected images from FRGC. DICTA includes the training images used to design the templates, DICTB has images of the same subject corresponding to the templates but

TABLE 3: KNOWN KEY LENGTH: IRFR (in %) with (0,1,2) bit ECC

# of	Lights			Nolights			Multi PIE			Palmprint			FRGC		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	0.38	0.00	0.00	1.64	0.00	0.00	0.19	0.02	0.01	10.00	0.48	0.11	14.47	0.57	0.14
10	1.21	0.00	0.00	2.91	0.00	0.00	0.46	0.07	0.03	18.65	1.05	0.32	30.08	2.11	0.53
20	1.97	0.00	0.00	4.37	0.00	0.00	1.07	0.25	0.17	31.23	2.65	1.14	59.15	6.70	1.68
30	2.42	0.00	0.00	5.74	0.00	0.00	1.86	0.61	0.47	40.40	4.12	2.35	80.36	14.68	4.39
40	3.48	0.00	0.00	7.37	0.64	0.46	3.21	1.34	1.11	47.38	5.89	3.90	92.96	25.92	9.49
50	4.00	0.00	0.00	8.56	1.18	1.18	4.88	2.50	2.24	53.18	8.00	5.86	97.88	36.94	15.63
60	4.84	0.00	0.00	11.08	3.09	2.27	7.32	4.40	4.04	58.23	10.34	8.32	99.62	51.92	26.89
70	5.38	0.03	0.03	13.57	4.83	4.67	9.94	6.60	6.10	62.41	13.20	11.15	99.95	64.56	39.11
80	6.21	0.08	0.08	17.66	8.32	7.52	13.00	9.30	8.78	65.87	15.26	13.49	100.0	76.26	53.31

TABLE 4: KNOWN KEY LENGTH: FCIRR WITHOUT RANDOM MASK (in %) with (0,1,2) bit ECC

# of	Lights			Nolights			Multi PIE			Palmprint			FRGC		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	13.30	13.30	13.30	9.16	9.16	9.16	22.42	22.42	22.42	0.44	0.44	0.44	27.81	27.81	27.81
10	2.78	2.78	2.78	1.13	1.13	1.13	9.99	9.99	9.99	0.02	0.02	0.02	13.76	13.76	13.76
20	0.21	0.21	0.21	0.03	0.03	0.03	2.84	2.84	2.84	0.00	0.00	0.00	4.60	4.60	4.60
30	0.02	0.02	0.02	0.00	0.00	0.00	0.76	0.76	0.76	0.00	0.00	0.00	0.88	0.88	0.88
40	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.19	0.19	0.00	0.00	0.00	0.11	0.11	0.11
50	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.03	0.00	0.00	0.00	0.02	0.02	0.02
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00



(a) Multi PIE (known) (b) Palmprint (known) (c) Multi PIE (unknown) (d) Palmprint (unknown)

Fig. 6: Information retrieval failure rate (in %) with known key size (a,b) and with unknown key size (c,d).

TABLE 5: UNKNOWN KEY LENGTH: IRFR (in %) with (0,1,2) bit ECC

# of	Lights			Nolights			Multi PIE			Palmprint			FRGC		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	0.38	0.00	0.00	1.64	0.00	0.00	0.32	0.14	0.13	10.04	0.53	0.18	14.47	0.66	0.25
10	1.21	0.00	0.00	2.91	0.00	0.00	0.59	0.22	0.19	18.65	1.15	0.53	30.21	2.46	1.11
20	1.97	0.00	0.00	4.37	0.00	0.00	1.32	0.57	0.52	31.29	3.11	1.98	59.69	9.39	5.90
30	2.42	0.00	0.00	5.92	0.46	0.46	2.57	1.54	1.47	40.50	5.26	4.12	81.49	25.63	20.68
40	3.48	0.00	0.00	7.83	2.19	2.19	4.73	3.31	3.23	47.52	8.15	7.12	93.93	53.05	49.01
50	4.00	0.00	0.00	9.75	4.28	4.28	7.64	6.04	5.95	53.36	11.70	10.67	98.66	78.48	76.40
60	4.84	0.00	0.00	14.18	8.25	8.08	11.73	9.92	9.79	58.49	15.64	14.84	99.83	93.38	92.76
70	5.53	0.26	0.26	19.39	14.40	14.30	15.92	14.07	13.92	62.80	20.14	19.47	99.99	98.51	98.36
80	6.36	0.67	0.67	25.68	20.33	20.26	20.84	18.94	18.80	66.31	24.12	23.68	100.0	99.74	99.68

under a different illumination setting and DICTC does not contain any images corresponding to the template subjects. The dictionary attack is performed by solving Eq. 23 for each template for every value of shift in the image (i.e., all possible values of Γ) to estimate an image maximizing the matching score. This estimated image is then used to retrieve the key from the corresponding template. In the latter scenario, we were unable to retrieve correct secret key bound to any of filters designed for the experiments described above since the random mask is unknown to the adversary. Table 7 lists the key-retrieval rates in the former scenario averaged over 10 different runs. As expected the quality of the dictionary determines the success rate of such dictionary attacks. Interestingly the efficacy of dictionary attacks also depends on the number of peaks (and hence key sizes bound to the template). Dictionary attacks against templates bound with longer keys result in poorer image estimates which in turn lead to poorer key retrieval rates.

7 Discussion

In this section we discuss some extensions to the general framework and some limitations of our algorithm.

7.1 Spreading Keys over Multiple Patterns

The ability to spread the keys over multiple pattern classes is a very powerful feature to have since it opens up biometric key-binding to many different security configurations. Notable examples of such applications are,

TABLE 6: UNKNOWN KEY LENGTH: FCIRR WITHOUT RANDOM MASK (in %) with (0,1,2) bit ECC

# of	Lights			Nolights			Multi PIE			Palmprint			FRGC		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	2.85	2.85	2.85	0.98	0.98	0.98	4.83	4.83	4.83	0.02	0.02	0.02	23.18	23.18	23.18
10	0.22	0.22	0.22	0.03	0.03	0.03	0.97	0.97	0.97	0.00	0.00	0.02	7.58	7.58	7.58
20	0.01	0.01	0.01	0.00	0.03	0.03	0.07	0.07	0.07	0.00	0.00	0.00	1.06	1.06	1.06
30	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.05	0.05	0.05
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

TABLE 7: DICTIONARY ATTACK: Key Retreival Rate (in %) with (0,1,2) bit ECC

#	Unknown Key Size									Known Key Size								
of	Nolights			Lights			FRGC			Nolights			Lights			FRGC		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	99.0	99.0	99.0	81.0	81.0	81.0	52.8	52.8	52.8	99.0	99.3	99.3	92.5	92.5	93.4	94.1	94.1	94.1
10	92.5	92.5	92.5	35.4	35.4	35.4	4.92	4.92	4.92	92.5	92.5	92.5	70.8	70.8	71.1	58.0	58.0	58.7
20	68.5	68.5	68.5	0.32	0.32	0.32	0.00	0.00	0.00	71.1	71.1	71.8	20.3	20.3	20.3	5.57	5.57	5.57
30	38.0	38.0	38.0	0.00	0.00	0.00	0.00	0.00	0.00	45.2	45.2	45.2	2.95	2.95	2.95	0.33	0.33	0.33
40	10.8	10.8	10.8	0.00	0.00	0.00	0.00	0.00	0.00	20.3	20.3	20.7	0.33	0.33	0.33	0.00	0.00	0.00
50	1.60	1.60	1.60	0.00	0.00	0.00	0.00	0.00	0.00	6.89	6.89	7.21	0.00	0.00	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.64	1.64	1.64	0.00	0.00	0.00	0.00	0.00	0.00
70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.33	0.00	0.00	0.00	0.00	0.00	0.00
80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.33	0.00	0.00	0.00	0.00	0.00	0.00

TABLE 8: Multi-Class: IRFR (in %) with (0,1,2) bit ECC

#	Unknown Key Size									Known Key Size								
of	Lights			Nolights			Palmprint			Lights			Nolights			Palmprint		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
05	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0
20	0.0	0.0	0.0	0.1	0.0	0.0	0.5	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.4	0.0	0.0
30	0.0	0.0	0.0	0.3	0.1	0.1	0.4	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.4	0.0	0.0
40	0.0	0.0	0.0	0.5	0.4	0.3	0.8	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.7	0.0	0.0
50	0.1	0.0	0.0	1.4	1.2	1.0	0.7	0.0	0.0	0.1	0.0	0.0	0.2	0.1	0.1	0.7	0.0	0.0
60	0.2	0.0	0.0	1.4	1.1	1.0	2.9	2.0	1.7	0.1	0.0	0.0	0.2	0.1	0.1	1.0	0.2	0.1
70	0.3	0.1	0.1	2.8	2.3	2.0	1.6	0.8	0.6	0.2	0.0	0.0	0.4	0.1	0.1	0.8	0.1	0.0
80	0.1	0.0	0.0	3.6	3.0	2.6	3.9	3.0	2.5	0.1	0.0	0.0	0.5	0.2	0.2	1.1	0.3	0.3

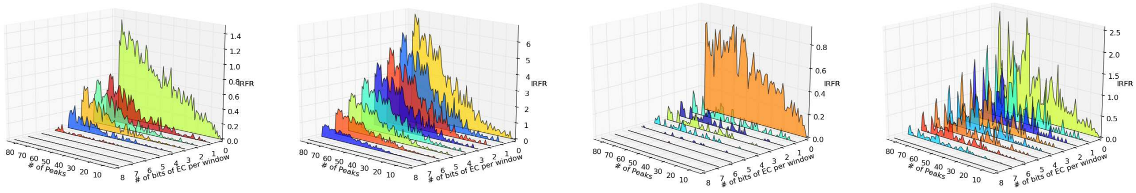
TABLE 9: Multi-Modal: IRFR (in %) with (0,1,2) bit ECC

#	Unknown Key Size						Known Key Size					
of	Lights			Nolights			Lights			Nolights		
Peaks	0	1	2	0	1	2	0	1	2	0	1	2
05	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
10	0.4	0.2	0.1	0.4	0.2	0.2	0.2	0.0	0.0	0.2	0.0	0.0
20	0.2	0.0	0.0	0.4	0.2	0.1	0.2	0.0	0.0	0.3	0.0	0.0
30	0.4	0.0	0.0	1.3	1.0	0.8	0.4	0.0	0.0	0.4	0.1	0.1
40	1.0	0.5	0.4	2.8	2.2	1.8	0.5	0.0	0.0	0.6	0.2	0.1
50	1.0	0.5	0.4	2.3	1.8	1.5	0.6	0.0	0.0	0.5	0.1	0.0
60	2.0	1.4	1.2	3.7	2.9	2.5	0.6	0.1	0.1	0.7	0.2	0.1
70	0.9	0.3	0.3	5.6	4.7	4.0	0.5	0.0	0.0	1.1	0.5	0.4
80	0.5	0.0	0.0	6.1	5.1	4.4	0.5	0.0	0.0	1.2	0.6	0.5

- Spreading keys across multiple people (could be different biometric modalities) can be useful for banking or similar applications. This would also be useful in scenarios where one would like to spread keys over n users but only require any $k(< n)$ user biometric samples to successfully retrieve the whole key.
- Spreading keys across multiple biometric modalities like face, palmprint, iris etc. thus making biometric authentication systems more resistant to social attacks.

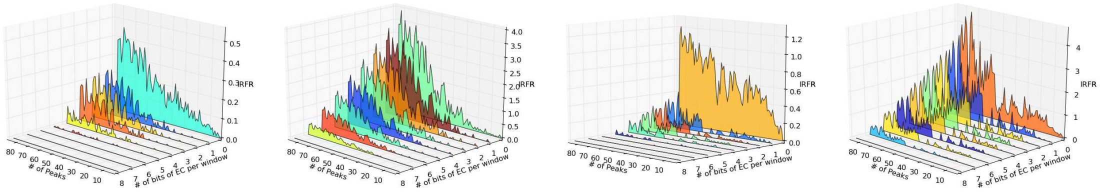
One can easily spread the key over multiple pattern classes by creating separate templates for each pattern class. However designing a single template such that query patterns from the constituent pattern classes can each retrieve a part of the bound key which can then be put together to retrieve the whole key, makes the information bound to the template more secure and also has lower storage space requirements.

Here we consider the problem of spreading a key over different biometric modalities (classes) while still designing a single template. We design the template by simply assigning the same class label while formulating the filter design with each training image having a shift determined by the part of the key that is bound to that training image. In our experiments we spread the key equally between the two modalities (classes), however one can spread the key



(a) Known Key Size (b) Unknown Key Size (c) Known Key Size (d) Unknown Key Size

Fig. 7: Information retrieval failure rate (in %) for multi-modal combination of PALMPRINT with (a,b) PIE NOLIGHTS and (c,d) PIE LIGHTS.



(a) Known Key Size (b) Unknown Key Size (c) Known Key Size (d) Unknown Key Size

Fig. 8: Information retrieval failure rate (in %) for multi-class combination (a,b) CMU PIE NOLIGHTS and (c,d) POLYU PALMPRINT

unevenly between the two as well.

For the multi-modal experiments we use faces and palmprints as the two biometric modalities over which we spread the keys. We generate random pairings of faces and palmprints for our experiments since to the best of our knowledge there is no publicly available multi-modal database of faces and palmprints. Further since there is no intrinsic relation in the images between faces and palmprints except that both belong to the same person, it is reasonable to create random pairs of faces from CMU-PIE and palmprints from the PolyU database. For the multi-class experiments we randomly pair users within each database considered, while ensuring that the pairings are unique.

We report the average results of each experiment conducted 10 times with a random key bound to the template and random pairing for the multi-modal and multi-class experiments for each run. We apply the filter individually to the two biometric signatures, extract the keys from each correlation output and combine the two to extract the complete key. Since an error in the extracted key can be due to either a single biometric signature or both, the accuracy of the key would depend on the specific combination of the biometric signatures among the test images, so we consider all possible pairs to report our results.

Fig. 7 shows the plots of IRFR for the multi-modal experiments when the key size is known and unknown. Similarly, Fig. 8 shows plots of IRFR for the multi-class experiments when the key size is known and unknown. The key-retrieval rates fluctuate a lot unlike the experiments in Section 6 due to the sensitivity of the algorithm to the class pairs i.e., some image pairs are easier than other pairs for key retrieval. This seems to suggest that given a choice for forming class pairs, one has to choose the pairs more carefully to optimize for IRFR. Table 9 and Table 8 list the performance for a select number of peaks with known and unknown key size for different multi-modal and multi-class database combinations respectively. Observe that the IRFRs for the multi-class and multi-modal case are much lower in comparison with the single biometric scenario for the same key length. The IRFR performance depends on the amount of noise in the correlation plane which in turn depends on the number of peaks in the correlation plane. Since the number of peaks per image are fewer the IRFR performance improves for these scenarios. So, in theory spreading the key over a greater number of pattern classes will improve the key retrieval rates for the same key size. However, more constraints in the filter optimization problem reduces the filter solution space resulting in a worse objective value. From these results we conclude that the bottleneck to binding longer keys to the template is the ability to detect the peaks in correlation output as opposed to the number of constraints. This trade-off that exists between the two warrants a more careful study and is a topic for future research. The FCIRR for the mutli-modal and multi-class settings is observed to be zero when using the secondary input and follows trends similar to Table 4 and Table 6 when the secondary password is not used i.e., the images are not convolved with the random mask.

7.2 Limitations

The main limitation of the proposed scheme is its limited robustness to large appearance variations as is evident from the FRGC results. We believe that using CFs with more distortion tolerance can help improve the algorithm's tolerance to larger image variations. In addition, as is evident from section 6.3.4, the proposed technique has limited robustness against dictionary attacks, at least for short keys, when not using a password to convolve the training images with a random mask. Designing classifiers which are tolerant to dictionary attacks can help increase robustness to such attacks and is a topic for future research.

8 Conclusions

We discussed a framework to bind information to image patterns and to retrieve this information during authentication by embedding the information in the template designed to discriminate that pattern class from the other pattern classes. We showed that we can robustly retrieve keys up to 250 bits with a information retrieval failure rate (IRFR) under 6% and nearly-zero false class information retrieval rate (FCIRR). We further showed how the same framework can be used to spread keys over multiple users and over multiple biometric modalities and retrieve keys up to 800 bits long with a IRFR of under 4% and nearly-zero FCIRR.

Acknowledgments

This research was supported by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office.

References

[1] D. Maltoni, *Handbook of fingerprint recognition*. Springer-Verlag New York Inc, 2003.

[2] V. Boddeti, F. Su, and B. V. K. Vijaya Kumar, "A biometric key-binding and template protection framework using correlation filters," *Advances in Biometrics*, pp. 919–929, 2009.

[3] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. K. Vijaya Kumar, "Biometric encryption using image processing," in *Proceedings of SPIE*, vol. 3314, 1998, p. 178.

[4] ——, "Biometric encryption: enrollment and verification procedures," in *Proceedings of SPIE*, vol. 3386, 1998, p. 24.

[5] ——, "Biometric encryption," *ICSA Guide to Cryptography*, pp. 649–675, 1999.

[6] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[7] U. Uludag, S. Pankanti, and A. Jain, "Fuzzy vault for fingerprints," in *Audio-and Video-Based Biometric Person Authentication*. Springer, 2005, pp. 310–319.

[8] Y. Lee, K. Park, S. Lee, K. Bae, and J. Kim, "A new method for generating an invariant iris private key based on the fuzzy vault system," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 5, pp. 1302–1313, 2008.

[9] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Transactions on Computers*, pp. 1081–1088, 2006.

[10] Y. Sutcu, S. Rane, J. Yedidia, S. Draper, and A. Vetro, "Feature extraction for a slepian-wolf biometric system using ldpc codes," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*. IEEE, 2008, pp. 2297–2301.

[11] ——, "Feature transformation of biometric templates for secure biometric systems based on error correcting codes," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, pp. 1–6.

[12] W. Scheirer and T. Boult, "Cracking fuzzy vaults and biometric encryption," in *Biometrics Symposium, 2007*. IEEE, pp. 1–6.

[13] A. Kholmatov and B. Yanikoglu, "Realization of correlation attack against the fuzzy vault scheme," in *Proc. of SPIE Symposium on Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819. Citeseer.

[14] K. Nandakumar, A. Nagar, and A. Jain, "Hardening fingerprint fuzzy vault using password," *Advances in Biometrics*, pp. 927–937, 2007.

[15] A. Teoh, A. Goh, and D. Ngo, "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1892–1901, 2006.

[16] B. V. K. Vijaya Kumar, A. Mahalanobis, and R. Juday, *Correlation pattern recognition*. Cambridge Univ Pr, 2005.

[17] A. Mahalanobis, B. V. K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Applied Optics*, vol. 26, no. 17, pp. 3633–3640, 1987.

[18] A. Mahalanobis, B. V. K. Vijaya Kumar, S. Song, S. Sims, and J. Epperson, "Unconstrained correlation filters," *Applied Optics*, vol. 33, no. 17, pp. 3751–3759, 1994.

[19] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550.

[20] N. Ratha, S. Chikkerur, J. Connell, and R. Bolle, "Generating cancelable fingerprint templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 561–572, 2007.

[21] M. Savvides, B. V. K. Vijaya Kumar, and P. Khosla, "Cancelable biometric filters for face recognition," *Pattern Recognition*, vol. 3, pp. 922–925, 2004.

[22] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha, "Secure and robust iris recognition using random projections and sparse representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1877–1893, 2011.

[23] D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[24] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[25] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1999.

[26] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009.

[27] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1615–1618, 2003.

[28] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

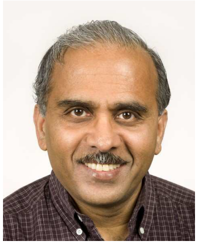
[29] D. Zhang, "Polyu palmprint palmprint database," *Biometric Research Centre, Hong Kong Polytechnic University. Available online at http://www. comp. polyu. edu. hk/˜ biometrics*.

[30] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1. IEEE, 2005, pp. 947–954.

[31] P. Hennings-Yeomans, B. V. K. Vijaya Kumar, and M. Savvides, "Palmprint classification using multiple advanced correlation filters and palm-specific segmentation," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 3, pp. 613–622, 2007.

[32] B. V. K. Vijaya Kumar, D. Carlson, and A. Mahalanobis, "Optimal trade-off synthetic discriminant function filters for arbitrary devices," *Optics Letters*, vol. 19, no. 19, pp. 1556–1558, 1994.

Vishnu Naresh Boddeti received a BTech degree in Electrical Engineering from the Indian Institute of Technology, Madras in 2007. He is currently in the Electrical and Computer Engineering program at Carnegie Mellon University where he received the MSc degree and is a candidate for the PhD degree. His research interests are in Computer Vision, Pattern Recognition and Machine Learning.



B.V.K. Vijaya Kumar is a Professor in the ECE Department at Carnegie Mellon University. His research interests include Pattern Recognition and Coding and Signal Processing for Data Storage Systems. He has co-authored one book, 15 book chapters and about 500 technical papers. Professor Kumar is a Fellow of IEEE, SPIE, OSA and IAPR. In 2003, he received the Eta Kappa Nu award for Excellence in Teaching in the ECE Department at CMU and the Carnegie Institute of Technology's Dowd Fellowship for educational contributions and in 2009, he was a co-recipient of the Carnegie Institute of Technology Outstanding Faculty Research award.